

Even $1 \times n$ Edge-Matching and Jigsaw Puzzles are Really Hard

Jeffrey Bosboom¹, Erik D. Demaine¹, Martin L. Demaine¹, Adam Hesterberg¹,
Pasin Manurangsi^{*2}, and Anak Yodpinyanee^{†1}

¹Massachusetts Institute of Technology, Cambridge, MA 02139, USA
jbosboom, edemaine, mdemaine, achester, anak@mit.edu

²University of California, Berkeley, CA 94720, USA
pasin@berkeley.edu

Abstract

We prove the computational intractability of rotating and placing n square tiles into a $1 \times n$ array such that adjacent tiles are compatible—either equal edge colors, as in edge-matching puzzles, or matching tab/pocket shapes, as in jigsaw puzzles. Beyond basic NP-hardness, we prove that it is NP-hard even to approximately maximize the number of placed tiles (allowing blanks), while satisfying the compatibility constraint between nonblank tiles, within a factor of 0.9999999851. (On the other hand, there is an easy $\frac{1}{2}$ -approximation.) This is the first (correct) proof of inapproximability for edge-matching and jigsaw puzzles. Along the way, we prove NP-hardness of distinguishing, for a directed graph on n nodes, between having a Hamiltonian path (length $n - 1$) and having at most $0.999999284(n - 1)$ edges that form a vertex-disjoint union of paths. We use this gap hardness and gap-preserving reductions to establish similar gap hardness for $1 \times n$ jigsaw and edge-matching puzzles.

1 Introduction

Jigsaw puzzles [Wil04] and edge-matching puzzles [Hau95] are two ancient types of puzzle, going back to the 1760s and 1890s, respectively. Jigsaw puzzles involve fitting together a given set of pieces (usually via translation and rotation) into a desired shape (usually a rectangle), often revealing a known image or pattern. The pieces are typically squares with a pocket cut out of or a tab attached to each side, except for boundary pieces which have one flat side and corner pieces which have two flat sides. Most jigsaw puzzles have unique tab/pocket pairs that fit together, but we consider the generalization to “ambiguous mates” where multiple tabs and pockets have the same shape and are thus compatible.

Edge-matching puzzles are similar to jigsaw puzzles: they too feature square tiles, but instead of pockets or tabs, each edge has a color or pattern. In *signed* edge-matching puzzles, the edge labels come in complementary pairs (e.g., the head and tail halves of a colored lizard), and adjacent tiles must have complementary edge labels on their shared edge (e.g., forming an entire lizard of one color). This puzzle type is essentially identical to jigsaw puzzles, where complementary pairs

^{*}Part of this work was completed while the author was at MIT and Dropbox, Inc.

[†]Research supported by NSF grant CCF-1420692.

of edge labels act as identically shaped tab/pocket pairs. In *unsigned* edge-matching puzzles, edge labels are arbitrary, and the requirement is that adjacent tiles must have identical edge labels. In both cases, the goal is to place (via translation and rotation) the tiles into a target shape, typically a rectangle.

A recent popular (unsigned) edge-matching puzzle is *Eternity II* [Wik16], which featured a US\$2,000,000 prize for the first solver (before 2011). The puzzle remains unsolved (except presumably by its creator, Christopher Monckton). The best partial solution to date [Thu09] either places 247 out of the 256 pieces without error, or places all 256 pieces while correctly matching 467 out of 480 edges.

Previous work. The first study of jigsaw and edge-matching puzzles from a computational complexity perspective proved NP-hardness [DD07]. Four years later, unsigned edge-matching puzzles were proved NP-hard even for a target shape of a $1 \times n$ rectangle [EFW11]. There is a simple reduction from unsigned edge-matching puzzles to signed edge-matching/jigsaw puzzles [DD07], which expands the puzzle by a factor of two in each dimension, thereby establishing NP-hardness of $2 \times n$ jigsaw puzzles. Unsigned $2 \times n$ edge-matching puzzles were claimed to be APX-hard (implying they have no PTAS) [AL10], but the proof is incorrect.¹

Our results. We prove that $1 \times n$ jigsaw puzzles and $1 \times n$ edge-matching puzzles are both NP-hard, even to approximate within a factor of 0.9999999851 ($> \frac{67038719}{67038720}$). This is the first correct inapproximability result for either problem. Even NP-hardness is new for $1 \times n$ signed edge-matching/jigsaw puzzles. By a known reduction [DD07], these results imply NP-hardness for polyomino packing (exact packing of a given set of polyominoes into a given rectangle) when the polyominoes all have area $\Theta(\log n)$; the previous NP-hardness proof [DD07] needed polyominoes of area $\Theta(\log^2 n)$.

We prove inapproximability for two different optimization versions of the problems. First, we consider placing the maximum number of tiles without any violations of the matching constraints. This objective has a simple $\frac{1}{2}$ -approximation for $1 \times n$ puzzles: alternate between placing a tile and leaving a blank. (A $\frac{2}{3}$ -approximation is also possible; see Section 7.) Second, we consider placing all of the tiles while maximizing the number of compatible edges between adjacent tiles (as in [AL10]).² This objective also has a simple $\frac{1}{2}$ -approximation for $1 \times n$ puzzles, via a maximum-cardinality matching on the tiles in a graph where edges represent having any compatible edges: any solution with k compatibilities induces a matching of size at least $k/2$ [AL10]. Thus, up to constant factors, we resolve the approximability of these puzzles.

Our reduction is from Hamiltonian path on directed graphs whose vertices each has maximum in-degree and out-degree 2, which was shown to be NP-complete by Plesník [Ple79]. To prove inapproximability, we reduce from a maximization version of this Hamiltonicity problem, called *maximum vertex-disjoint path cover*, where the goal is to choose as many edges as possible to form vertex-disjoint paths. (A Hamiltonian path would be an ideal path cover, forming a single path of length $|V| - 1$.) This problem has a $\frac{12}{17}$ -approximation [Vis92], and is NP-hard to approximate within some constant factor [Eng03] via a known connection to Asymmetric TSP with weights in $\{1, 2\}$ [Vis92].

¹Personal communication with Antonios Antoniadis, October 2014. In particular, Lemma 3’s proof is incomplete.

²A dual objective would be to place all tiles while minimizing the number of mismatched edges, but this problem is already NP-hard to distinguish between an answer of zero and positive, so it cannot be approximated.

We prove that maximum vertex-disjoint path cover satisfies a stronger type of hardness, called *gap hardness*: it is NP-hard to distinguish between a directed graph having a Hamiltonian path versus one where all vertex-disjoint path covers having at most $0.999999284|V|$ ($> \frac{1396639}{1396640}|V|$) edges, given a promise that the graph falls into one of these two categories. This gap hardness immediately implies inapproximability within a factor of 0.999999284 (though this constant is weaker than the known inapproximability bound [Eng03]). More useful is that our reduction to $1 \times n$ jigsaw/edge-matching puzzles is gap-preserving, implying gap hardness and inapproximability for the latter. This approach lets us focus on “perfect” instances (where all tiles are compatible) versus “very bad” instances (where many tiles are incompatible), which seems far easier than standard L-reductions used in many inapproximability results, where we must distinguish between an arbitrary optimal and a factor below that arbitrary optimal. We posit that gap hardness—where the high end of the gap is “perfect”, attaining the maximum possible bound, and thus matching the NP-hardness of the original decision problem—is the better way to prove hardness of approximation for many puzzles and games, and hope that our results and approach find use in other research as well.

2 Problem Definitions

In this section, we formally define the relevant concepts and problems we consider in this paper.

2.1 Approximation Algorithms and Gap Problems

One approach to dealing with NP-hard problems is to allow an algorithm to approximate the answer to the problem instead of solving it exactly. The quality of such an *approximation algorithm* is given by its *approximation ratio* as defined below. For the purposes of this paper, we define approximation ratio and subsequent notions only for maximization problems; these notions can be extended for minimization problems as well.

Definition 1. *An approximation algorithm for a maximization problem has an approximation ratio $\rho \leq 1$ if it outputs a solution that has value at least ρ times the optimum. We call such an algorithm a ρ -approximation algorithm.*

Whereas many NP-hard problems admit polynomial-time ρ -approximation algorithms for some sensible ρ 's, some are proven to be NP-hard to approximate to within some approximation ratio. To prove such inapproximability results, one typically reduces from a known NP-hard problem (e.g., 3SAT) to a maximization problem so that, if the original instance is a YES instance, then the optimum of the resulting instance is large and, if the original instance is a no instance, then the optimum is small. This polynomial-time reduction implies that the optimization is NP-hard to approximate to within the ratio between the optimums in the two cases. For convenience, we use the following notation of *gap problems* to describe these inapproximability results.

Definition 2. *For a maximization problem P and $\beta > \gamma$, $\text{GAP } P[\beta, \gamma]$ is the problem of distinguishing whether an instance of P has optimum at least β or at most γ .*

Gap problems are widely used in hardness of approximation because NP-hardness of $\text{GAP } P[\beta, \gamma]$ (which we refer to informally as *gap hardness*) implies NP-hardness of approximating P to within a factor of γ/β .

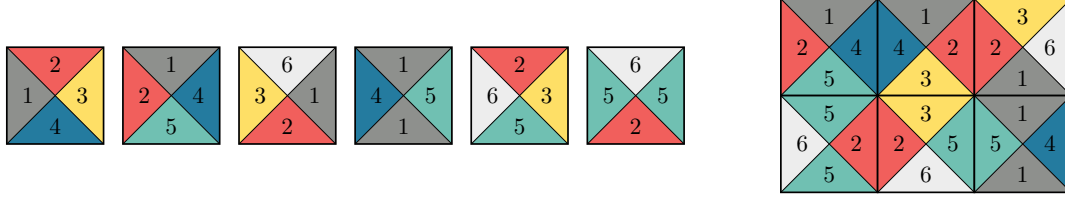


Figure 1: An example of a 2×3 EDGE-MATCHING PUZZLE instance consisting of 6 tiles (left), where all tiles can be assembled into a 2×3 rectangular grid, with matching colors on the edges of adjacent tiles (right). Colors are specified redundantly as numbers.

2.2 Edge-Matching and Jigsaw Puzzles

Next, we give formal definitions of our main problems of interest, (unsigned) edge-matching and signed edge-matching puzzles (or jigsaw puzzles), as described in the introduction. While our paper will focus on the $1 \times n$ rectangular grid case, we define the more general problem of edge-matching puzzles on an $h \times w$ rectangular grid. We begin with the unsigned decision problem:

Definition 3. $h \times w$ EDGE-MATCHING PUZZLE is the following problem:

INPUT: $n = hw$ unit-square tiles where each of the four sides has a color.

OUTPUT: whether the tiles can be assembled into the $h \times w$ rectangular grid (board) so that any two adjacent pieces have the same color on the shared edge.

Figure 1 shows an example of a 2×3 EDGE-MATCHING PUZZLE. We write $\mathcal{T}(c_1, c_2, c_3, c_4)$ to denote a tile with colors c_1, c_2, c_3, c_4 on its left, upper, right, and lower edges respectively. For example, the first tile in Figure 1 is $\mathcal{T}(1, 2, 3, 4)$.

In an assembly, we allow tiles to be rotated but not reflected. It is easy to see from the proofs that all of our results hold if we allow reflections in addition to rotations. On the other hand, if rotations are prohibited, then $1 \times n$ EDGE-MATCHING PUZZLE is in P [EFW11]: with this restriction, the problem becomes equivalent to finding an Eulerian path (a path that visits every edge exactly once) in the directed graph whose vertices represent the colors and whose edges represent the tiles, which can be solved in linear time.

Next, we define two optimization versions of the edge-matching problem. In the first version, the goal is to put as many tiles in the grid without any shared edge being labeled with different colors on two tiles. In the second version (previously considered in [AL10]), we must place all n tiles into the rectangular grid, and the goal is to maximize the total number of matching edges between adjacent tiles.

Definition 4. $h \times w$ MAX-PLACEMENT EDGE-MATCHING PUZZLE is the following problem:

INPUT: $n = hw$ unit-square tiles where each of the four sides has a color.

OUTPUT: the maximum number of the tiles that can be assembled³ into the $h \times w$ rectangular grid such that any two adjacent pieces have the same color on the shared edge.

Definition 5. $h \times w$ MAX-MATCHED EDGE-MATCHING PUZZLE is the following problem:

INPUT: $n = hw$ unit-square tiles where each of the four sides has a color and a sign on it.

³To avoid confusion, we clarify that each tile must be placed parallel to the board's borders with integer-coordinated corners, i.e., the board has hw unit-square slots and each tile is either put in a slot or left out of the board.

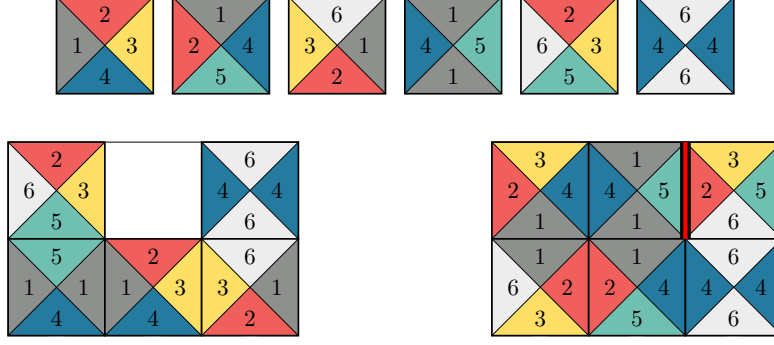


Figure 2: An instance (top) corresponding to both a 2×3 MAX-PLACEMENT EDGE-MATCHING PUZZLE whose optimal tiling contains 5 tiles (bottom left), and a 2×3 MAX-MATCHED EDGE-MATCHING PUZZLE whose optimal tiling yields 6 matched edges (bottom right); the mismatched edge is indicated with a thick line.

OUTPUT: the maximum number of same-color edges between adjacent tiles obtained by assembling all n tiles into $h \times w$ rectangular grid.

Figure 2 shows an example comparing both optimization problems. In this paper, we focus on proving inapproximability of $1 \times n$ Max-Placement Edge-Matching Puzzle, then later show how to modify the proof to establish hardness under the Max-Matched variation.

Next we define the *signed* variation of the puzzle, where each edge has a sign and the signs on the shared edge of two pieces must be opposite.

Definition 6. $h \times w$ SIGNED EDGE-MATCHING PUZZLE is the following problem:

INPUT: $n = hw$ unit-square tiles where each of the four sides has a color and a sign (+ or -).

OUTPUT: whether the tiles can be assembled into the $h \times w$ rectangular grid so that any two adjacent pieces have the same color but opposite signs on the shared edge.

Likewise, we use $\mathcal{T}(s_1c_1, s_2c_2, s_3c_3, s_4c_4)$ to denote a tile with colors c_1, c_2, c_3, c_4 and signs s_1, s_2, s_3, s_4 on its left, upper, right, and lower edges respectively.

Signed edge-matching puzzles can be viewed as jigsaw puzzles: + and - signs correspond to tabs and pockets respectively. It is worth noting, however, that the definition above is slightly different from typical jigsaw puzzles: signed edge-matching puzzles allow tabs and pocket on the borders, whereas jigsaw puzzles normally forbid this. If the borders of the puzzle must be flat (have no color or sign), then $1 \times n$ jigsaw puzzles become solvable in polynomial time in a similar fashion to edge-matching puzzles without rotations.

For signed edge-matching puzzles, we can define an optimization version similarly:

Definition 7. $h \times w$ MAX-PLACEMENT SIGNED EDGE-MATCHING PUZZLE is the following problem:

INPUT: $n = hw$ unit-square tiles where each of the four sides has a color and a sign on it.

OUTPUT: the maximum number of the tiles that can be assembled into the $h \times w$ rectangular grid such that any two adjacent pieces have the same color and different signs on the shared edge.

It is possible to define the Max-Matched variation for signed edge-matching puzzles, and our inapproximability result still holds. Note, however, that this variation is unnatural when interpreted

as a jigsaw puzzle, because it would be physically difficult to place mismatched jigsaw pieces adjacent to each other.

2.3 Hamiltonian Path and Maximum Vertex-Disjoint Path Cover

The Hamiltonian cycle problem is to decide whether a given (directed) graph has a *Hamiltonian cycle*—a cycle that visits each vertex in the graph exactly once. This problem has long been known to be NP-complete; it was on Karp’s initial list of 21 NP-complete problems [Kar72]. In this paper, we are more interested in its close relative, the Hamiltonian path problem:

Definition 8. HAMILTONIAN PATH *is the following problem:*

INPUT: *a directed graph $G = (V, E)$, a source s , and a sink t .*⁴

OUTPUT: *whether there exists a Hamiltonian path (a path visiting each vertex exactly once) starting at s and ending at t .*

Like the Hamiltonian cycle problem, the Hamiltonian path problem has long been known to be NP-complete [GJ79].⁵ Garey, Johnson, and Tarjan [GJT76] showed that the problem remains NP-hard even on undirected graphs where each vertex has degree at most three. Plesník [Ple79] extended the result to planar directed graphs with bounded in- and out-degrees.

While we do not use Plesník’s result directly, we use his reduction from 3SAT to prove gap hardness of $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE. To do so, we need an optimization version of the Hamiltonian path problem:

Definition 9. MAX VERTEX-DISJOINT PATH COVER(d) *is the following problem:*

INPUT: *a directed graph $G = (V, E)$ with a unique source s and a unique sink t such that the in-degree and out-degree of each vertex are at most d .*

OUTPUT: *the maximum number of edges in any vertex-disjoint path cover—a collection of paths such that each vertex appears in exactly one path. Here we allow paths of length zero (i.e., a single vertex).*

MAX VERTEX-DISJOINT PATH COVER(d) can be viewed as an optimization version of the Hamiltonian path problem because a graph has a vertex-disjoint path cover with $|V| - 1$ edges if and only if there exists a Hamiltonian path. The uniquenesses of source and sink and the degree requirement are imposed for technical reasons, which will be clear once our reduction is presented.

Engebretsen [Eng03] proved that Asymmetric TSP with weights in $\{1, 2\}$ is NP-hard to approximate within a factor of $\frac{2805}{2804} - \varepsilon$, which implies [Vis92] that MAX VERTEX-DISJOINT PATH COVER(d) is NP-hard to approximate within some constant factor (better than the constant we will obtain). Their proof implies a gap hardness result, but not one where the high end of the gap has perfect solutions (optimal value $|V| - 1$, i.e., Hamiltonian), at least not without modification. To fix this problem, and for self-containedness, we analyze our own construction here.

⁴A *source* of a directed graph is a vertex with in-degree zero; a *sink* is a vertex with out-degree zero.

⁵The conventional definition of HAMILTONIAN PATH (e.g., in [GJ79]) does not require s and t to be a source and a sink, respectively. We may easily reduce the conventional problem to our variant by adding a new starting vertex s' with a single incident edge from s' to s , and a new target vertex t' with a single incident edge from t to t' . Any Hamiltonian path in one instance can be converted into a corresponding Hamiltonian path in the other instance by adding or removing these new vertices and edges, establishing NP-hardness result for our variant.

2.4 MAX-3SAT

To establish gap hardness for MAX VERTEX-DISJOINT PATH COVER(d), we resort to a classic problem: MAX-3SAT. Recall that a Boolean formula is in *conjunctive normal form (CNF)* if it is a conjunction (AND) of zero or more clauses, where each clause is a disjunction (OR) of at most three literals, each of which is either a variable or its negation. For example, $(x_1 \vee x_2) \wedge (x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee x_4)$ is in CNF with three clauses.

The celebrated Cook's Theorem (also known as Cook-Levin Theorem) states that it is NP-hard to decide whether a CNF formula is satisfiable [Coo71]. Shortly after, Karp [Kar72] proved NP-completeness of the more restricted version 3SAT, where each clause has at most three literals. Because we are aiming for an inapproximability result, we will use an optimization version of 3SAT:

Definition 10. MAX-3SAT(k) is the following problem:

INPUT: a CNF formula such that each clause contains at most three literals and each variable appears in at most k clauses.

OUTPUT: the maximum number of clauses satisfied by any assignment.

The PCP Theorem, considered a landmark in modern complexity theory, essentially states that MAX-3SAT (without the bound k on variable occurrences) is NP-hard to approximate to within some constant factor [ALM⁺98, AS98]. MAX-3SAT(k) has also been researched intensively; even before the PCP Theorem was proven, Papadimitriou and Yannakakis [PY88] showed that, for some constant k , MAX-3SAT(k) is a complete problem for a complexity class MaxSNP. Later, Feige [Fei98] proved that MAX-3SAT(5) is NP-hard to approximate up to some constant factor. For the purpose of this paper, we will use the following gap hardness of MAX-3SAT(29) from [Vaz01, pp. 314], which has a more concrete ratio than that from [Fei98].

Lemma 1. For any constant $\alpha_{3\text{SAT}} < \frac{1}{344}$, GAP MAX-3SAT(29)[$m, (1 - \alpha_{3\text{SAT}})m$] is NP-hard.

3 Our Contributions

With formal problem statements in hand, we now formally state our results, starting with our main result for $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE.

Theorem 1. For any nonnegative constant $\alpha_{\text{EMP}} < \frac{1}{67038720}$, GAP $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE[$n, (1 - \alpha_{\text{EMP}})n$] is NP-hard. In particular, it is NP-hard to approximate $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE to within a factor of $\frac{67038719}{67038720} + \delta$ of optimal for any sufficiently small constant $\delta > 0$.

Although NP-hardness of $h \times w$ EDGE-MATCHING PUZZLE is known for every $h \geq 1$ [EFW11], we are not aware of any existing inapproximability result for edge-matching puzzles. Our proof for the above theorem also yields an analogous result for $1 \times n$ MAX-PLACEMENT SIGNED EDGE-MATCHING PUZZLE as stated below.

Corollary 1. For any nonnegative constant $\alpha_{\text{EMP}} < \frac{1}{67038720}$, GAP $1 \times n$ MAX-PLACEMENT SIGNED EDGE-MATCHING PUZZLE[$n, (1 - \alpha_{\text{EMP}})n$] is NP-hard. In particular, it is NP-hard to approximate $1 \times n$ MAX-PLACEMENT SIGNED EDGE-MATCHING PUZZLE to within a factor of $\frac{67038719}{67038720} + \delta$ of optimal for any sufficiently small constant $\delta > 0$.

In addition, we prove a similar inapproximability result for the variation considered in [AL10] as stated below. Note that this result also holds for its corresponding signed variation by simply applying the proofs of both corollaries.

Corollary 2. *For any nonnegative constant $\alpha_{\text{EMP}} < \frac{1}{67038720}$, GAP $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE $[n, (1 - \alpha_{\text{EMP}})n]$ is NP-hard. In particular, it is NP-hard to approximate $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE to within a factor of $\frac{67038719}{67038720} + \delta$ of optimal for any sufficiently small constant $\delta > 0$.*

As an intermediate step to showing the results above, we prove gap hardness for MAX VERTEX-DISJOINT PATH COVER(2):

Theorem 2. *For any nonnegative constant $\alpha_{\text{MPC}} < \frac{1}{1396640}$, GAP MAX VERTEX-DISJOINT PATH COVER(2) $[|V| - 1, (1 - \alpha_{\text{MPC}})(|V| - 1)]$ is NP-hard. In particular, it is NP-hard to approximate MAX VERTEX-DISJOINT PATH COVER(2) to within a factor of $\frac{1396639}{1396640} + \delta$ of optimal for any sufficiently small constant $\delta > 0$.*

The outline for the rest of the paper is as follows. In Section 4, we prove NP-hardness of $1 \times n$ SIGNED EDGE-MATCHING PUZZLE by a reduction from HAMILTONIAN PATH; this is not one of the main results described above, yet its simple proof provides good intuition for the subsequent inapproximability proofs. Next, in Section 5, we prove the inapproximability results for $1 \times n$ edge-matching puzzles, Theorem 1 and Corollaries 1–2, using a similar reduction from MAX VERTEX-DISJOINT PATH COVER(2), an intermediate problem in achieving our results. We then prove the inapproximability result for MAX VERTEX-DISJOINT PATH COVER(2) in Section 6. Finally, we discuss further potential directions for research for the edge-matching problem in Section 7.

4 NP-hardness of $1 \times n$ Signed Edge-Matching Puzzle

In this section we give a simple proof of NP-hardness of $1 \times n$ SIGNED EDGE-MATCHING PUZZLE. We provide a polynomial-time reduction from HAMILTONIAN PATH to $1 \times n$ SIGNED EDGE-MATCHING PUZZLE in Section 4.1. We then prove in Section 4.2 that our reduction satisfies the following lemma:

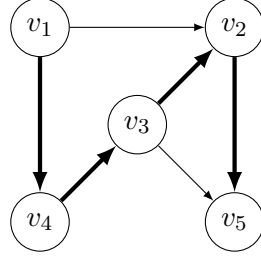
Lemma 2. *The polynomial-time reduction from HAMILTONIAN PATH to $1 \times n$ SIGNED EDGE-MATCHING PUZZLE described in Section 4.1 is such that the HAMILTONIAN PATH instance contains a Hamiltonian path if and only if the tiles of the constructed $1 \times n$ SIGNED EDGE-MATCHING PUZZLE instance can be assembled into an $1 \times n$ grid without any mismatched edges.*

This lemma and NP-hardness of HAMILTONIAN PATH imply our desired hardness result:

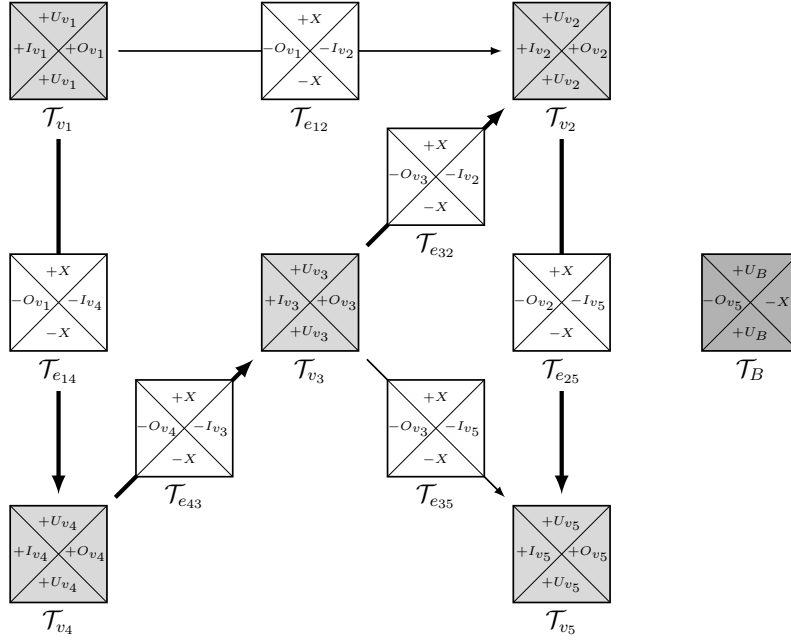
Corollary 3. *$1 \times n$ SIGNED EDGE-MATCHING PUZZLE is NP-hard.*

4.1 Construction of $1 \times n$ SIGNED EDGE-MATCHING PUZZLE Instance from HAMILTONIAN PATH Instance

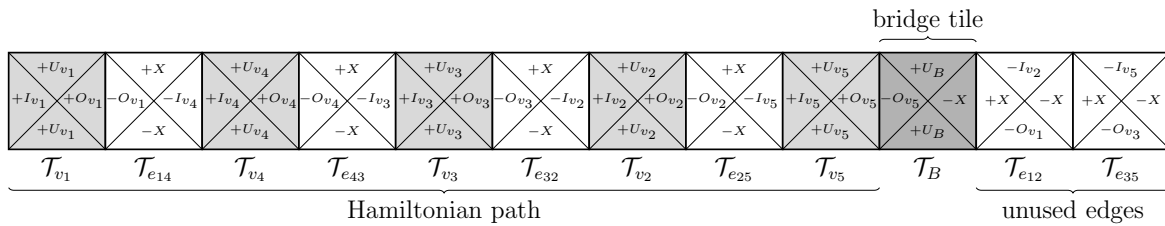
The overall strategy of the reduction from HAMILTONIAN PATH to $1 \times n$ SIGNED EDGE-MATCHING PUZZLE is to represent each vertex and each edge by tiles, encoding the path in the ordering of the tiles. The leftover edge tiles will be packed at the end of the tiling. The path and the leftover edges in our tiling are then joined by a bridge tile.



(a) An example of a HAMILTONIAN PATH instance, a directed graph G with source $s = v_1$ and sink $t = v_5$. There exists a Hamiltonian path from s to t , shown with thick edges.



(b) The constructed $1 \times n$ SIGNED EDGE-MATCHING PUZZLE instance (with $n = 12$). On the graph structure are the vertex tiles (light gray) and the edge tiles (white). The only bridge tile (dark gray) is given on the right.



(c) The tiling constructed from the Hamiltonian path above. The path is represented in the first nine slots; observe that each slot contains a tile along the path in the above figure. The tenth slot contains the bridge tile, whereas the last two are filled with unused edge tiles sharing garbage-colored edges.

Figure 3: An example of the reduction described in Section 4.1, together with a Hamiltonian path of the HAMILTONIAN PATH instance, and the corresponding tiling without any mismatched edges in the constructed $1 \times n$ SIGNED EDGE-MATCHING PUZZLE instance.

The reduction, illustrated in Figure 3a–3b, takes an instance of HAMILTONIAN PATH, which is a directed graph $G = (V, E)$ with a source s and a sink t , then produces an instance of $1 \times n$ SIGNED EDGE-MATCHING PUZZLE as follows:

- For each vertex $v_i \in V$, we add a *vertex tile* \mathcal{T}_{v_i} with positive *vertex colors* $+I_{v_i}$ and $+O_{v_i}$ on opposite edges (each represents entering and leaving v_i , respectively), and the unique unmatchable color $+U_{v_i}$ on the remaining edges; i.e., $\mathcal{T}_{v_i} = \mathcal{T}(+I_{v_i}, +U_{v_i}, +O_{v_i}, +U_{v_i})$.
- For each edge $e_{ij} \in E$ from v_i to v_j , we add an *edge tile* $\mathcal{T}_{e_{ij}}$ with negative vertex colors $-O_{v_i}$ and $-I_{v_j}$ on opposite edges (each represents leaving v_i and entering v_j , respectively), and the matching *garbage colors* pair $+X$ and $-X$ on the two remaining edges. In other words, $\mathcal{T}_{e_{ij}} = \mathcal{T}(-O_{v_i}, +X, -I_{v_j}, -X)$.
- Finally, we have one *bridge tile* \mathcal{T}_B with the negative vertex color $-O_t$ on one edge, one garbage color $-X$ on the opposite edge, and a unique unmatchable color $+U_B$ on the remaining edges. That is, $\mathcal{T}_B = \mathcal{T}(-O_t, +U_B, -X, +U_B)$.

4.2 Proof of Lemma 2

Our reduction is clearly a polynomial-time reduction. We must now show that the existence of a Hamiltonian path in the HAMILTONIAN PATH instance implies the existence of a tiling scheme without any mismatched edges in the constructed $1 \times n$ SIGNED EDGE-MATCHING PUZZLE instance, and vice versa.

HAMILTONIAN PATH $\implies 1 \times n$ SIGNED EDGE-MATCHING PUZZLE

Turning a Hamiltonian path into a tiling scheme is easy; an example of this process can be found in Figure 3c. We encode our path into the leftmost $2|V| - 1$ slots in the grid by placing the tiles corresponding to the vertices and edges in the path alternately in the grid. More specifically, let the Hamiltonian path be $v_{\pi(1)}, \dots, v_{\pi(|V|)}$ where $v_{\pi(1)} = s$ and $v_{\pi(|V|)} = t$. For $j = 1, \dots, |V|$, we place the vertex tile \mathcal{T}_{v_j} oriented as $\mathcal{T}(+I_{v_{\pi(j)}}, X, +O_{v_{\pi(j)}}, X)$ in the $(2j - 1)^{\text{th}}$ slot. For $j = 1, \dots, |V| - 1$, we place the edge tile $\mathcal{T}_{e_{\pi(j)\pi(j+1)}}$ oriented as $\mathcal{T}(-O_{v_{\pi(j)}}, +X, -I_{v_{\pi(j+1)}}, -X)$ in the $(2j)^{\text{th}}$ slot.

We then place the bridge tile oriented as $\mathcal{T}_B = \mathcal{T}(-O_t, +U_B, -X, +U_B)$ in the $(2|V|)^{\text{th}}$ slot; since t is the last vertex in the Hamiltonian path, this tile is compatible with the last tile in the $(2|V| - 1)^{\text{th}}$ slot. For the remaining edge tiles, we rotate them as $\mathcal{T}(+X, -I_{v_j}, -X, -O_{v_i})$ so that the matching garbage colors are on the left and right, then put them into the rest of the slots. Clearly, we have placed every tile on the board without any conflict and, hence, we have completed the first half of the proof.

$1 \times n$ SIGNED EDGE-MATCHING PUZZLE \implies HAMILTONIAN PATH

Assume that there exists a tiling scheme on a $1 \times n$ grid without any mismatched edges. Observe that the bridge tile must be oriented so that its unique color $+U_B$ is on its upper and lower edges because this color cannot be matched by any tile. Without loss of generality, suppose that the bridge tile is oriented as $\mathcal{T}_B = \mathcal{T}(-O_t, +U_B, -X, +U_B)$. (Otherwise, rotate the whole grid 180° .)

Consider first the tiles on the right side of the bridge tile. By our construction, we may only place edge tiles oriented as $\mathcal{T}(+X, -I_{v_j}, -X, -O_{v_i})$ in order to match their garbage-colored

edges. Consequently, none of the vertex tiles may be placed; all of them must appear on the left side of the bridge tile.

Now consider the tiles on the left side of the bridge tile. Observe that, to be compatible with the color $-O_{v_j}$ to the right, we may only place the vertex tile \mathcal{T}_{v_j} oriented as $\mathcal{T}(+I_{v_j}, +U_{v_j}, +O_{v_j}, +U_{v_j})$. Similarly, to be compatible with the color $+I_{v_j}$ to the right, we may only place some edge tile $\mathcal{T}_{e_{ij}}$ oriented as $\mathcal{T}(-O_{v_i}, +X, -I_{v_j}, -X)$. Therefore, these tiles may only be placed if they correspond to a path on G ending at t . Since all vertex tiles appear to the left of the bridge tile, we have a path visiting all vertices ending at t . Moreover, as our starting vertex s is a source, the color $+I_s$ cannot be matched by any tile. Thus \mathcal{T}_s must be the left-most tile on the grid. That is, the tiles to the left of the bridge tile encode a Hamiltonian path of G , as desired.

5 Inapproximability of $1 \times n$ Edge-Matching Puzzles

In this section, we generalize the approach from Section 4 in order to prove Theorem 1, NP-hardness of approximation of $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE. To this end, we use the optimization variant of HAMILTONIAN PATH called MAX VERTEX-DISJOINT PATH COVER defined earlier, and provide a gap-preserving reduction from MAX VERTEX-DISJOINT PATH COVER to $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE. We largely focus on the unsigned variant because it is the more complicated case. Specifically, by removing the signs in the reduction of Section 4.1, it becomes possible to place two edge tiles next to each other without using the garbage color since they share an additional color if they have the same starting vertex or ending vertex. We circumvent this problem by restricting to graphs of in-degrees and out-degrees at most two. This requires us to later show the hardness of approximation for MAX VERTEX-DISJOINT PATH COVER(2) in this more restricted family of graphs.

We describe our reduction from MAX VERTEX-DISJOINT PATH COVER(2) to $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE in Section 5.1. We then prove in Section 5.2 that the reduction satisfies the properties stated in the lemma below:

Lemma 3. *For any nonnegative constants α_{MPC} and $\alpha_{\text{EMP}} < \alpha_{\text{MPC}}/48$, the following properties hold for the reduction described in Section 5.1 when $|V|$ is sufficiently large:*

- *if the optimum of the MAX VERTEX-DISJOINT PATH COVER(2) instance is $|V| - 1$ (the graph contains a Hamiltonian path), then the optimum of the resulting $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE instance is at least n , i.e., every tile can be placed compatibly on the board, and,*
- *if the optimum of the MAX VERTEX-DISJOINT PATH COVER(2) instance is at most $(1 - \alpha_{\text{MPC}})(|V| - 1)$, then the optimum of the resulting $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE is at most $(1 - \alpha_{\text{EMP}})n$.*

We will later show NP-hardness of approximation of MAX VERTEX-DISJOINT PATH COVER(2), namely Theorem 2, in Section 6. The above lemma and Theorem 2 immediately imply Theorem 1.

Then, we describe how to modify the reduction to arrive at a similar inapproximability result for $1 \times n$ MAX-PLACEMENT SIGNED EDGE-MATCHING PUZZLE (Corollary 1) in Section 5.3. Lastly, we provide another reduction to obtain the inapproximability result for $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE (Corollary 2) in Section 5.4.

5.1 Construction of $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE Instance from MAX VERTEX-DISJOINT PATH COVER(2) Instance

The overall strategy of the reduction from MAX VERTEX-DISJOINT PATH COVER(2) to $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE remains unchanged from Section 4.1, except that the signs on the tiles are now removed, as shown in Figure 4. For completeness, we include a concise specification below:

- For each vertex $v_i \in V$, add a vertex tile $\mathcal{T}_{v_i} = \mathcal{T}(I_{v_i}, U_{v_i}, O_{v_i}, U_{v_i})$.
- For each vertex $e_{ij} \in E$, add an edge tile $\mathcal{T}_{e_{ij}} = \mathcal{T}(O_{v_i}, X, I_{v_j}, X)$.
- Add a bridge tile $\mathcal{T}_B = \mathcal{T}(O_t, U_B, X, U_B)$.

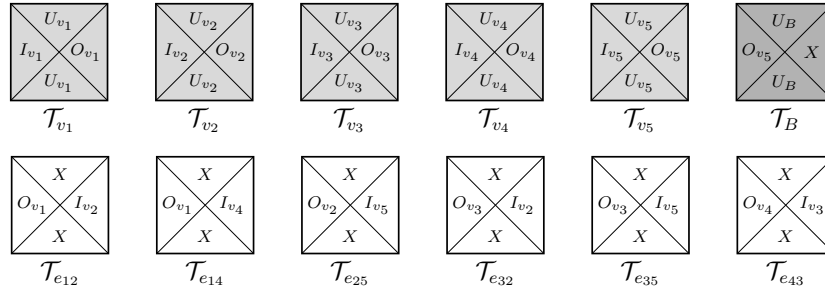


Figure 4: The tiles of the $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE instance created by the reduction in Section 5.1, where the MAX VERTEX-DISJOINT PATH COVER(2) instance is the same graph from Figure 3a.

Turning a vertex-disjoint path cover into a tiling scheme is still straightforward. We can simply encode each path by placing the tiles corresponding to the vertices and edges in the path alternately in the grid, leaving spaces between paths as necessary. Finally, we arrange the remaining edge tiles together by sharing garbage-colored edges. Intuitively, if there are only few paths in the path cover, then the number of blank spaces on the board is also small. In particular, if there is a Hamiltonian path, then all tiles can still be placed without any mismatched edges.

On the other hand, converting a tiling configuration to a path cover is not as easy. Ideally, we want every edge tile $\mathcal{T}_{e_{ij}}$ that is packed in the grid to fall into one of the following categories.

- It is oriented so that the vertex-colored edges are on the left and the right ($\mathcal{T}(O_{v_i}, X, I_{v_j}, X)$ or $\mathcal{T}(I_{v_j}, X, O_{v_i}, X)$) and is placed between two vertex tiles. In this case, the corresponding edge should be included in the path cover.
- It is oriented so that the garbage colors are on the left and right (i.e., $\mathcal{T}(X, I_{v_j}, X, O_{v_i})$ or $\mathcal{T}(X, O_{v_i}, X, I_{v_j})$). These edges are discarded from the path cover.

Unfortunately, it is possible for two edge tiles to be placed next to each other without the shared border being labeled with the garbage color; for instance, if there are edges e_{12} and e_{13} , then we can rotate their corresponding tiles to be $\mathcal{T}(O_{v_2}, X, I_{v_1}, X)$ and $\mathcal{T}(I_{v_1}, X, O_{v_3}, X)$, and placed them next to each other. This is where the bounded degree constraint comes in. Suppose that each vertex in the graph has in-degree and out-degree at most two. If two edge tiles are placed consecutively

with vertex color on the shared edge, there must be an empty slot, blank space or the board's border next to the vertex tile corresponding to that color (if the vertex tile is placed on the board). As a result, if there are not too many empty slots, we can reason that there must also be a small number of such pairs of edge tiles. We can then remove these problematic edge tiles and arrive at a path cover as desired. These arguments are formalized below.

5.2 Proof of Lemma 3

We will prove each half of Lemma 3.

MAX VERTEX-DISJOINT PATH COVER(2) \implies $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE

This proof remains unchanged from that case of HAMILTONIAN PATH \implies $1 \times n$ SIGNED EDGE-MATCHING PUZZLE in Section 4.2, as removing the signs only makes the puzzle easier to solve.

$1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE \implies MAX VERTEX-DISJOINT PATH COVER(2)

We will prove this by contrapositive. Suppose that the resulting $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE instance has optimum more than $(1 - \alpha_{\text{EMP}})n$, i.e., there is a tiling with less than $\alpha_{\text{EMP}}n$ empty slots. We translate the tiling to a path cover as follows:

1. Remove the bridge tile from the board.
2. Consider all edge tiles that are rotated in such a way that left and right edges are vertex-colored. Among these tiles, remove the ones that are adjacent to a blank space, a (left or right) border of the board or another edge tile.
3. After the previous step, two consecutive edge tiles can only share a garbage-colored edge. Moreover, from the colors we choose for vertex tiles, two vertex tiles cannot be placed next to each other. As a result, each connected component (a maximal set of contiguous nonempty slots) either (1) alternates between vertex and edge tiles, or (2) consists solely of edge tiles that share only garbage-colored edges.

Because we removed all edge tiles with left and right vertex-colored edges that are next to a blank space or a board's border, each sequence of the first type must start and end with vertex tiles, which represents a path. We create a path cover consisting of all paths corresponding to such sequences, and length-zero paths, one for each vertex tile not in the board.

Figure 5 shows the algorithm in action for an example tiling scheme. Clearly, the resulting collection of paths is a vertex-disjoint path cover. We will now argue that this path cover indeed contains more than $(|V| - 1)(1 - \alpha_{\text{MPC}})$ edges. We start by proving the following useful lemma.

Lemma 4. *Less than $14\alpha_{\text{EMP}}n + 12$ edge tiles are removed in Step 2.*

Proof. After the bridge tile is removed, there are less than $\alpha_{\text{EMP}}n + 1$ empty slots. This means that, in Step 2, less than $2(\alpha_{\text{EMP}}n + 1) + 2 = 2\alpha_{\text{EMP}}n + 4$ edge tiles are removed because they are adjacent to a blank space or a border of the board. The rest must be removed because they share a vertex color-labeled edge with other edge tiles.

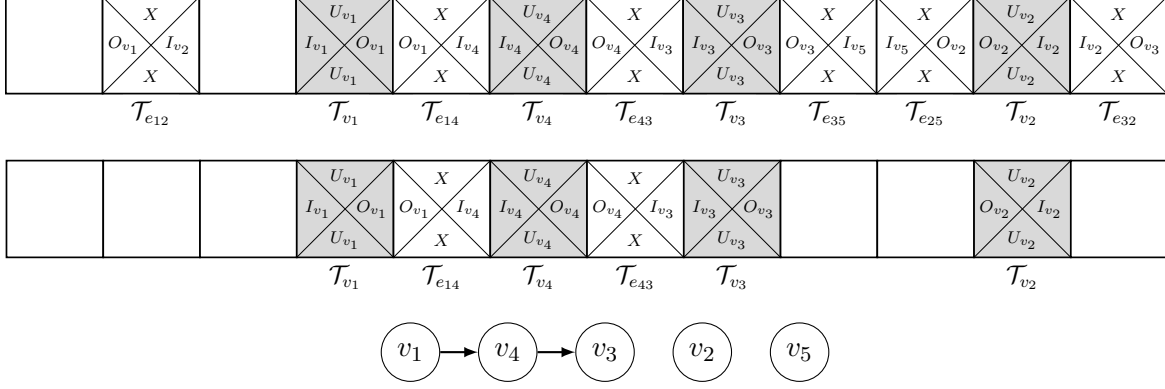


Figure 5: An example of how the algorithm works for a tiling scheme for the instance from Figure 4. The upper tiling is the input. Since the bridge tile is not put on the board, the tiling is not modified in Step 1. In Step 2, \mathcal{T}_{e12} and \mathcal{T}_{e32} are removed because they are next to a blank space and a board border respectively; \mathcal{T}_{e35} and \mathcal{T}_{e25} are removed because they share a vertex-colored edge. This results in the lower tiling. The third step of the algorithm then picks two paths $(v_1 \rightarrow v_4 \rightarrow v_3$ and $v_2)$ from the board and one (v_5) from an unused vertex tile.

Suppose that the shared edge of two consecutive edge tiles is labeled by I_{v_i} or O_{v_i} for some vertex v_i . Because v_i has in-degree and out-degree at most two, the tile \mathcal{T}_{v_i} must be either omitted from the board or is next to a blank space or a border of the board. There can be at most $\alpha_{\text{EMP}}n + 2\alpha_{\text{EMP}}n + 2 = 3\alpha_{\text{EMP}}n + 2$ such \mathcal{T}_{v_i} 's. For each such \mathcal{T}_{v_i} 's, there are only four edge tiles that have I_{v_i} or O_{v_i} . Hence, the number of edge tiles removed by such cause is at most $4(3\alpha_{\text{EMP}}n + 2) = 12\alpha_{\text{EMP}}n + 8$.

Thus, the total number of edge tiles removed is less than $14\alpha_{\text{EMP}}n + 12$. \square

Next, recall that the number of edges in the path cover is $|V|$ minus the number of paths. There are two type of paths we create: (1) paths from tiling in the board and (2) paths from unused vertex tiles. The number of the latter is bounded by the number of initially empty slots, which is less than $\alpha_{\text{EMP}}n$. The number of the former is at most the number of connected components in the board after Step 2, which is bounded by one plus the number of empty slots.

The empty slots on the board are either initially empty or are made empty in the first two steps. Recall that the number of initially empty slots is less than $\alpha_{\text{EMP}}n$. One additional empty slot occurs in Step 1. Finally, from Lemma 4, the number of empty slots created in Step 2 is less than $14\alpha_{\text{EMP}}n + 12$. Hence, the total number of empty slots is less than $15\alpha_{\text{EMP}}n + 13$. Thus, the number of edges in the path cover is more than

$$\begin{aligned} |V| - (1 + 15\alpha_{\text{EMP}}n + 13) - \alpha_{\text{EMP}}n &= |V| - 14 - 16\alpha_{\text{EMP}}(|V| + |E| + 1) \\ &\geq |V| - 14 - 16\alpha_{\text{EMP}}(|V| + 2|V| + 1) \\ &\geq (1 - 48\alpha_{\text{EMP}})|V| - 30. \end{aligned}$$

The first inequality follows from the fact that each vertex has in-degree at most two.

Finally, since $\alpha_{\text{EMP}} < \alpha_{\text{MPC}}/48$, when $|V|$ is sufficiently large, the last quantity is more than $(1 - \alpha_{\text{MPC}})(|V| - 1)$. We have thus completed the proof of Lemma 3.

5.3 Proof of Corollary 1 (Signed Variation)

Because the proof of Corollary 1 largely resembles that of Theorem 1, we will not repeat the whole proof here. The main difference is in the reduction: we instead need to follow the reduction from Section 4.1 by restoring the omitted signs. Apart from this, the rest of the proof remains unchanged.

5.4 Proof of Corollary 2 (Maximum-Matched Variation)

In this section, we provide a gap reduction to the maximum-matched variation, although this time our instance does not need to be modified. More specifically, we prove the following lemma.

Lemma 5. *For any constant $\alpha < 1$, the following properties hold:*

- *if the optimum of $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE is n , then the optimum of $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE on the same instance is $n - 1$, and,*
- *if the optimum of $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE is at most $(1 - \alpha)n$, then the optimum of $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE on the same instance is at most $(1 - \alpha)(n - 1)$.*

By picking $\alpha = \alpha_{\text{EMP}}$ from Theorem 1, Lemma 5 together with Theorem 1 immediately implies NP-hardness of GAP $1 \times n$ MAX-MATCHED EDGE-MATCHING PUZZLE $[n - 1, (1 - \alpha_{\text{EMP}})(n - 1)]$, establishing Corollary 2.

Proof of Lemma 5. For clarity, we refer to the $1 \times n$ MAX-PLACEMENT EDGE-MATCHING PUZZLE problem, where the goal is to maximize the number of matched edges, as the original problem.

Original problem \implies Maximum-Matched Variation

This is straightforward since a perfect tiling contains $n - 1$ matched edges.

Maximum-Matched Variation \implies Original problem

We again prove by contrapositive; suppose that the maximum matched edges variation has a solution with more than $(1 - \alpha)(n - 1)$ matched edges. That is, there are less than $\alpha(n - 1) < \alpha n$ mismatched edges. We create a solution to the original problem by removing all the tiles whose right edges are mismatched. Because each mismatched edge contributes to at most one empty slot, there are less than αn blank spaces, i.e., the number of tiles is more than $(1 - \alpha)n$. \square

We note that our proof holds for the signed variation, and so does our inapproximability result.

6 Inapproximability of MAX VERTEX-DISJOINT PATH COVER(2)

We now prove the inapproximability of MAX VERTEX-DISJOINT PATH COVER(2) via a reduction from MAX-3SAT(29). Throughout section, let n and m denote the number of variables and clauses in the CNF instance of MAX-3SAT(29), respectively.

Our proof of the inapproximability of MAX VERTEX-DISJOINT PATH COVER(2) primarily relies on the reduction used to prove NP-hardness of the Hamiltonian cycle problem on graphs with degree bound two in [Ple79]. We modify Plesník's construction to create a gap-preserving reduction, then

apply our reduction on MAX-3SAT(29). Section 6.1 addresses the construction in detail. This construction turns a CNF instance of MAX-3SAT(29) into a graph instance $G = (V, E)$ of MAX VERTEX-DISJOINT PATH COVER(2) satisfying the following lemma.

Lemma 6. *For any nonnegative constants $\alpha_{3\text{SAT}}$ and $\alpha_{\text{MPC}} = \alpha_{3\text{SAT}}/4060$, the following properties hold for the reduction described in Section 6.1:*

- *if the optimum of the MAX-3SAT(29) instance is m (the corresponding CNF formula is satisfiable), then the optimum of the resulting MAX VERTEX-DISJOINT PATH COVER(2) instance is $|V| - 1$ (the corresponding graph contains a Hamiltonian path), and,*
- *if the optimum of the MAX-3SAT(29) instance is at most $(1 - \alpha_{3\text{SAT}})m$, then the optimum of the resulting MAX VERTEX-DISJOINT PATH COVER(2) is at most $(1 - \alpha_{\text{MPC}})(|V| - 1)$.*

Our proof of Lemma 6 is provided in Section 6.2. Lemma 6 and Lemma 1 immediately imply the NP-hardness for the gap problem of MAX VERTEX-DISJOINT PATH COVER(2) described in Theorem 2.

6.1 Construction of MAX VERTEX-DISJOINT PATH COVER(2) Instance from MAX-3SAT(29) Instance

Assume that we are given a MAX-3SAT(29) instance ϕ ; namely, ϕ consists of n variables x_1, \dots, x_n , and m clauses c_1, \dots, c_m where each c_j is a disjunction of (at most) 3 literals $y_{j1}, y_{j2}, y_{j3} \in \{x_1, \overline{x_1}, \dots, x_n, \overline{x_n}\}$, such that no variable appears as literals more than 29 times. Using Plesník's construction, we similarly define the clause gadget and the variable gadget as shown in Figure 6a; each gadget is enclosed in a rounded dashed rectangle.⁶ Each variable gadget has two *literal edges* corresponding to x_i and $\overline{x_i}$, respectively. Figure 6b shows how we connect our gadgets together. Unlike Plesník's construction, we create a source node s and a target node t rather than creating an edge between the top vertices of the first clause and variable gadgets.

Each arc with black circular endpoints is a shorthand notation for an *exclusive-or line* (XOR line). The XOR gadget, which is the realization of an XOR line, is given in Figures 7a-7b. There is one XOR line within a variable gadget on the right, and an XOR line in the middle connecting a literal edge to each occurrence of that literal in each clause. In Figure 6a, for example, clause c_j contains a three literals, the middle of which is $y_{j2} = x_i$.

For each variable x_i , we define its *variable territory* as the set of vertices in its variable gadget, along with the eight vertices introduced by each of its XOR lines, including those on the clause gadgets where x_i or $\overline{x_i}$ appear. This is illustrated as the shaded region on Figure 6a. We also define a *clause territory* of each clause c_j as the union between the vertices of c_j 's variable gadget, and the variable territories of variables whose literals appear in c_j .

6.2 Proof of Lemma 6

We will next prove each half of Lemma 6.

⁶We contracted some edges in Plesník's gadgets; this allows us to achieve a slightly better value of α_{MPC} .

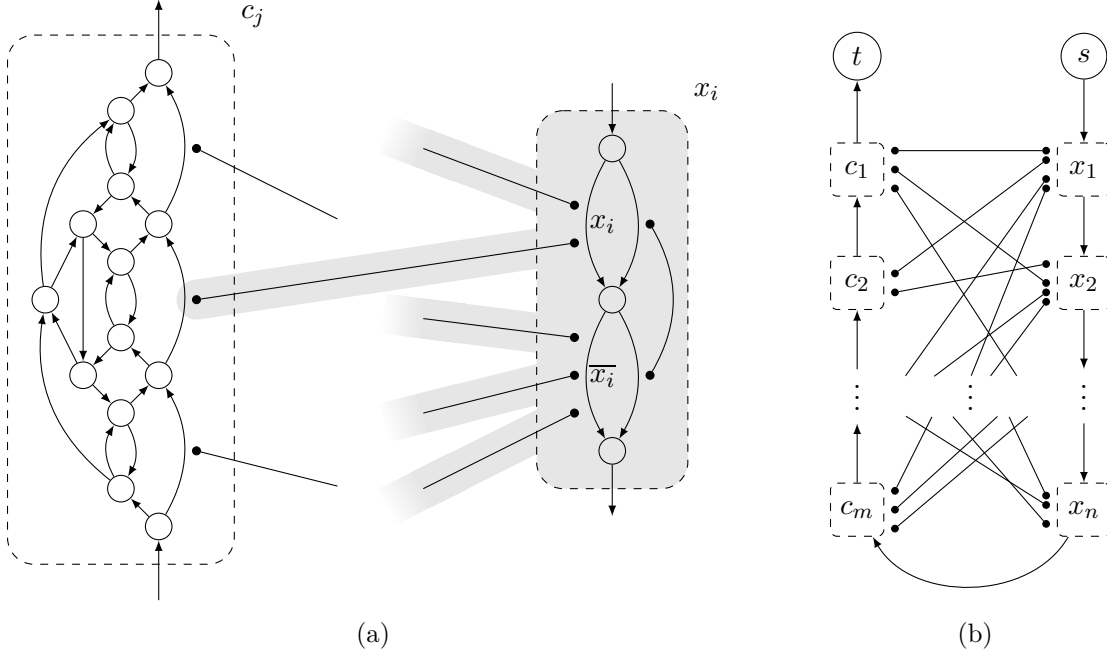


Figure 6: (a) A clause gadget for c_j (left) and a variable gadget for x_i (right), where c_j contains x_i as its second literal; the shaded area indicates x_i 's variable territory. (b) Full construction of our MAX VERTEX-DISJOINT PATH COVER(2) instance.

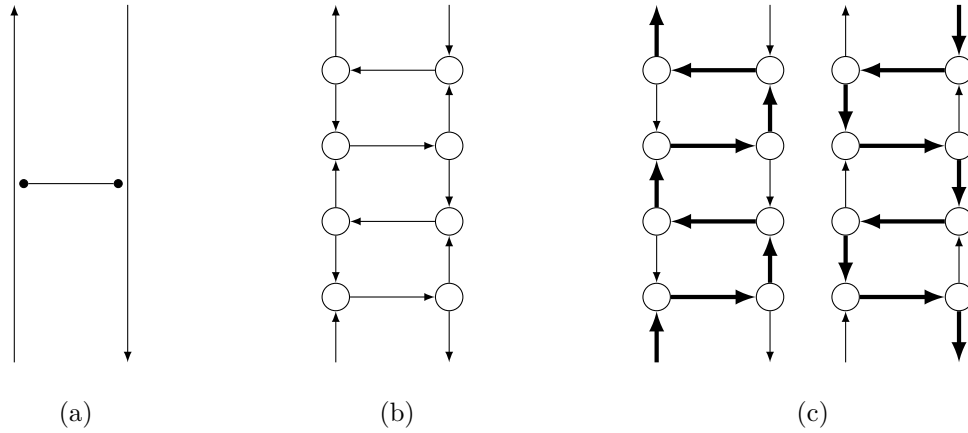


Figure 7: (a) The shorthand notation for an XOR line. (b) The XOR gadget realizing the XOR line of (a). (c) The only two possible configurations of path cover (thick edges) without inducing endpoints within an XOR line.

MAX-3SAT(29) \implies MAX VERTEX-DISJOINT PATH COVER(2)

Suppose that the input MAX-3SAT(29) instance is satisfiable. We need to show that the constructed MAX VERTEX-DISJOINT PATH COVER(2) instance contains a path cover of size $|V| - 1$. Observe that there is a path cover of size $|V| - 1$ if and only if the path cover consists entirely of a single (Hamiltonian) path from s to t .

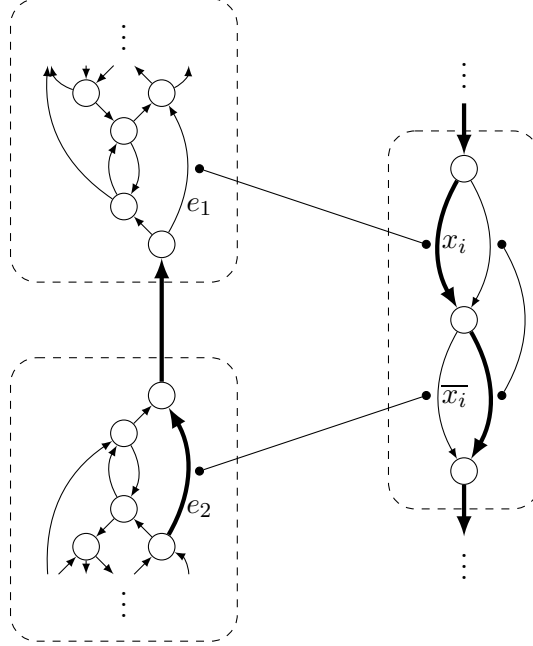


Figure 8: Thick edges indicate edges that must be in the path cover when x_i is set to TRUE. Edge e_1 corresponds to a TRUE literal (x_i), so it must not be traversed when we visit the clause side of the graph; edge e_2 corresponds to a FALSE literal ($\overline{x_i}$) and therefore must be traversed.

Let us define *endpoints* of a path cover as a set of the sources and the destinations of paths in the cover. Then, it is sufficient to give a path cover that has no other endpoints besides s and t . First, we establish the following observation which can be easily checked via a simple case analysis.

Observation 1. *Consider Figure 7c. If a path cover has no endpoints on any of the eight vertices introduced by an XOR line, then exactly one edge of the XOR line (in the shorthand notation) is in the path cover. More precisely, in the actual realization, either the pair of edges entering and leaving the XOR gadget on the left, or the pair on the right, is in the path cover.*

Now we are ready to explicitly provide the path cover; this is largely the same as that from in [Ple79], but included here for completeness. We begin by adding the literal edges corresponding to the optimal (satisfying) assignment of the MAX-3SAT(29); that is, if $x_i = \text{TRUE}$ we add the literal edge x_i to the cover; otherwise we add the literal edge $\overline{x_i}$. Due to Observation 1 above, the edges in the variable gadgets in our cover are fully determined; an example in case x_i is set to TRUE is given in Figure 8. Observe that the path starting from s visits all vertices of all variable gadgets, and, at the same time, restricts that the edge on the opposite end of the XOR line is traversed if and only if the corresponding literal is FALSE.

Now consider our path when it reaches the clause side of the graph. As shown in Figure 9a, if the clause is satisfied, some edges on the right of the clause gadget does not need to be traversed. By a simple case analysis, we can cover all unvisited vertices on the clause gadget without breaking up our path. Thus, we have constructed a path cover of size $|V| - 1$, as needed.

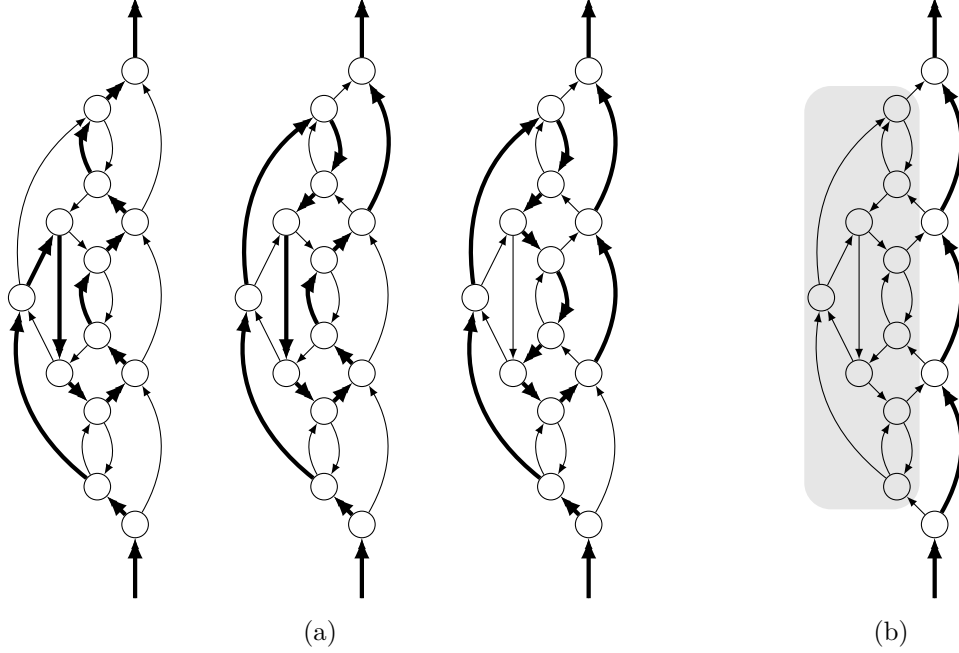


Figure 9: (a) Possible constructions of path covers when the clause is satisfied (symmetric cases are omitted). (b) The shaded region must contain an endpoint if the path cover includes all thick edges.

MAX VERTEX-DISJOINT PATH COVER(2) \implies MAX-3SAT(29)

We prove that second part of Lemma 6 via contrapositive. That is, suppose that the optimum of the constructed MAX VERTEX-DISJOINT PATH COVER(2) instance is greater than $(1 - \alpha_{\text{MPC}})(|V| - 1)$, we must show that the optimum of the original MAX-3SAT(29) instance is greater than $(1 - \alpha_{\text{3SAT}})m$.

Consider an optimal path cover of our constructed MAX VERTEX-DISJOINT PATH COVER instance. Our path cover is of size greater than $(1 - \alpha_{\text{MPC}})(|V| - 1)$, so it must contain less than $\alpha_{\text{MPC}}|V| + \alpha_{\text{MPC}} \leq \alpha_{\text{MPC}}|V| + 1$ paths. That is, besides s and t , our path cover contains strictly less than $2\alpha_{\text{MPC}}|V|$ other endpoints. To complete our proof, we will explicitly construct an assignment satisfying more than $(1 - \alpha_{\text{3SAT}})m$ clauses.

Now, observe that if a variable territory contains no endpoints, then exactly one of the two literal edges are in the path cover due to the XOR lines as argued in Observation 1. We construct our assignment as follows. If there are no endpoints in x_i 's variable territory, we assign the truth value according to the literal in the path cover. Otherwise, assign an arbitrary one. We now show that our assignment fails to satisfy strictly less than $\alpha_{\text{3SAT}}m$ clauses.

First, we prove the following lemma.

Lemma 7. *If a clause c_j is not satisfied by our assignment, then c_j 's clause territory contains an endpoint.*

Proof. Suppose that there are no endpoints in any of the variable territories of any variable occurring in c_j . Then for c_j to be unsatisfied, all literal edges in c_j must be FALSE and not included in the path cover. Observation 1 implies that we must traverse all three edges on the right of c_j 's clause

gadget; these are shown in Figure 9b as the curved thick edges. Notice that our path leaves the shaded vertices disconnected from the remaining part of the graph, which requires at least one additional path to cover. That is, there must still be an endpoint in the clause territory of c_j . \square

Lemma 7 implies that an endpoint must be present for a clause to be unsatisfied. Next, we show that each endpoint cannot be contained in very many clause territories.

Lemma 8. *Each endpoint is in at most 29 clause territories.*

Proof. If the endpoint is not in any variable territory, then it is in at most one clause territory because clause gadgets do not overlap. Otherwise, notice that the endpoint can be only in one variable territory as variable territories also do not overlap. One variable may occur at most 29 times, so a variable territory can be in at most 29 clause territories. Therefore, any endpoint can be in at most 29 clause territories. \square

Each endpoint may cause up to 29 clauses to become unsatisfied. Recall that there are less than $2\alpha_{\text{MPC}}|V|$ endpoints that belong to some clause territories. Thus our assignment leaves strictly less than $29 \cdot 2\alpha_{\text{MPC}}|V| = 58\alpha_{\text{MPC}}|V|$ clauses unsatisfied.

Finally, we relate $|V|$ and m in order to establish our claim. Notice that we add vertices s and t to our graph for convenience when defining endpoints, but they can be safely removed as their only neighbors do not have other incoming or outgoing edges, respectively. Then, the number of vertices becomes $|V| \leq 11n + 37m \leq 11(3m) + 37m = 70m$. As the number of unsatisfied clauses is less than $58\alpha_{\text{MPC}}(70m) = 4060m$, our claim holds because $\alpha_{\text{MPC}} = \alpha_{3\text{SAT}}/4060$, as desired.

7 Conclusion and Open Problems

Our results establish an upper bound on the optimal approximation factor at $\frac{67038719}{67038720}$, and the simple approximation algorithms mentioned in the Introduction give a lower bound of $\frac{1}{2}$. The optimal approximation factor remains an open question, though the right answer is probably closer to $\frac{1}{2}$. We believe our inapproximability factor can be tightened somewhat by further modifying Plesník's construction, or by reducing from a different problem; e.g., find a variation of SAT that yields a better trade-off in the reduction, or make use of the tighter bounds on $\{1, 2\}$ -TSP from [Eng03].

For the objective of placing the maximum tiles without violations, we can improve the approximation factor for $1 \times n$ puzzles to $\frac{2}{3}$ by computing a maximum-cardinality matching on the tiles, and interspersing matched groups (of size 2 or 1) with blanks. Suppose that an optimal solution contains $OPT = \lceil \frac{n}{2} \rceil + k$ tiles. Such a solution contains $n - (\lceil \frac{n}{2} \rceil + k) = \lfloor \frac{n}{2} \rfloor - k$ blank slots, leaving at least $(n - 1) - 2(\lfloor \frac{n}{2} \rfloor - k) \geq 2k - 1$ matched edges, inducing a matching of cardinality at least $\lceil \frac{2k-1}{2} \rceil = k$ for these matched edges which form paths. If $k \geq \frac{n}{3}$, this leads to an approximate solution consisting of pairs of matched tiles, except possibly for the last tile, packing at least $\frac{2}{3}n \geq \frac{2}{3}OPT$ tiles (because $OPT \leq n$). Otherwise, the solution contains $2k$ tiles in the first $3k$ slots, ending with a blank slot, followed by alternation in the remaining $n - 3k$ spaces, for a total of $2k + \lceil \frac{1}{2}(n - 3k) \rceil \geq \lceil \frac{n+k}{2} \rceil = \lceil \frac{OPT}{2} + \frac{\lfloor n/2 \rfloor}{2} \rceil \geq \frac{2}{3}OPT$ tiles placed.

Another natural question is whether our inapproximability results also apply to the original NP-hardness scenario of square target shapes [DD07]. We expect that our proofs can be adapted to this case, but it requires care to avoid introducing additional error in the approximation factor.

Edge-matching puzzles are also common with tiles in the shape of regular hexagons or equilateral triangles. Are these problems NP-hard and inapproximable for “ $1 \times n$ ” (but bumpy) target shapes? Our results carry over directly to hexagons, just by leaving two opposite edges of each hexagon unlabeled, and using the other two pairs of opposite edges to simulate a square tile. Two triangles connected by a unique glue can simulate a square, which suffices to prove NP-hardness, but more care would be required for inapproximability to consider the case that the triangle pair gets separated.

References

- [AL10] Antonios Antoniadis and Andrzej Lingas. Approximability of edge matching puzzles. In *Proceedings of the 36th Conference on Current Trends in Theory and Practice of Computer Science*, volume 5901 of *Lecture Notes in Computer Science*, pages 153–164, Špindlerův Mlýn, Czech Republic, January 2010.
- [ALM⁺98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, May 1998.
- [AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *Journal of the ACM*, 45(1):70–122, January 1998.
- [Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing*, STOC ’71, pages 151–158, New York, NY, USA, 1971. ACM.
- [DD07] Erik D. Demaine and Martin L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23 (Supplement):195–208, June 2007.
- [EFW11] Martin Ebbesen, Paul Fischer, and Carsten Witt. Edge-matching problems with rotations. In *Proceedings of the 18th International Symposium on Fundamentals of Computation Theory*, volume 6914 of *Lecture Notes in Computer Science*, pages 114–125, Oslo, Norway, August 2011.
- [Eng03] Lars Engebretsen. An explicit lower bound for TSP with distances one and two. *Algorithmica*, 35(4):301–319, 2003.
- [Fei98] Uriel Feige. A threshold of $\ln N$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, July 1998.
- [GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.
- [GJT76] Michael R. Garey, David S. Johnson, and Robert E. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976.
- [Hau95] Jacques Haubrich. *Compendium of Card Matching Puzzles*. Self-published, May 1995. Three volumes.

- [Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer US, 1972.
- [Ple79] Ján Plesník. The NP-completeness of the Hamiltonian cycle problem in planar diagraphs with degree bound two. *Information Processing Letters*, 8(4):199–201, April 1979.
- [PY88] Christos Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 229–234, New York, NY, USA, 1988. ACM.
- [Thu09] Christel Thunell. Lundafamilj bäst i världen på svårknäckt pussel. *Sydsvenskan*, January 21 2009. <http://www.sydsvenskan.se/lund/lundafamilj-bast-i-varlden-pa-svarknackt-pussel/>.
- [Vaz01] Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag New York, Inc., New York, NY, USA, 2001.
- [Vis92] Sundar Vishwanathan. An approximation algorithm for the asymmetric travelling salesman problem with distances one and two. *Information Processing Letters*, 44(6):297–302, December 1992.
- [Wik16] Wikipedia. Eternity II. https://en.wikipedia.org/wiki/Eternity_II_puzzle, January 2016.
- [Wil04] Anne D. Williams. *The Jigsaw Puzzle: Piecing Together a History*. Berkley Books, New York, 2004.